

DOCUMENT RESUME

ED 156 155

IR 006 137

AUTHOR Romaniuk, E. W.
TITLE Problems in Researching Course Adequacy.
PUB DATE Mar 78
NOTE 32p.; Paper presented at the Annual Meeting of the American Educational Research Association (Toronto, Canada, March 1978)

EDRS PRICE MF-\$0.83 HC-\$2.06 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; Course Evaluation; Editing; Educational Technology; Error Patterns; *Evaluation Methods; Field Studies; *Formative Evaluation; Measurement Techniques; Programed Instruction; Programing; Programing Problems

ABSTRACT

Without rigorous evaluation during the development cycle of computer assisted instruction programs, diagnosis and revision of the courseware is likely to be haphazard and inefficient, and the probability of the course being a success is greatly reduced. This article provides guidance for developers of computer assisted instructional software in performing formative evaluations of their programs. Techniques discussed include: (1) peer review of lesson scripts, (2) automatic computer support programs, (3) single student tests, (4) small group tests, (5) class tests, and (6) field tests. Diagnosis and revision of courseware is discussed with emphasis placed on systematic analysis of student performance on both the micro and macro levels. (Author/STS)

* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY.

PROBLEMS IN RESEARCHING COURSE ADEQUACY

BY

E.W. ROMANIUK

Division of Educational Research Services
The University of Alberta
EDMONTON, Alberta
T6G 2G5

"PERMISSION TO REPRODUCE THIS
MATERIAL HAS BEEN GRANTED BY

E. W. Romaniuk

TO THE EDUCATIONAL RESOURCES
INFORMATION CENTER (ERIC) AND
USERS OF THE ERIC SYSTEM."

A paper to be presented at the
AERA Annual Meeting
March, 1978

ABSTRACT

A great deal of literature in the CAI field has focused upon the hardware and software requirements for CAI systems and authoring languages. Emphasis has been upon development of appropriate student terminals and easy to use, but powerful authoring languages. In addition, numerous studies have examined the efficiency of CAI versus conventional classroom instruction. However, one CAI component appears to be lacking in the literature. That component is the search for sets of formalized evaluation and optimization procedures to increase the probability of production of quality courseware. More emphasis is required in the area of systematic formative courseware evaluation and courseware optimization procedures. Formal evaluation review procedures can begin early in the courseware development cycle by having the author's peers examine CAI scripts prior to entry into the CAI system. As the courseware development cycle nears completion, evaluation activities would necessarily intensify and diversify. Many of the decisions related to diagnosis of courseware adequacy and instructional areas in need of revision can only be done on the basis of the student performance. To aid the author in this process, a number of computer support programs are described. Provision of quick and easy access to various student summary reports should ease the time consuming aspect of the formative evaluation and revision process and encourage authors to optimize their courseware.

INTRODUCTION

Many CAI authors often assume that once they have designed and developed their desired courseware, most of the difficult and time consuming work has been done. Sure, they have studied the numerous flowcharts associated with the various models for courseware development and realize there is a component called "evaluation" still to be done. However, evaluation must be a very minor component of the total design and implementation cycle, especially when compared to some of the difficulties in trying to adhere to rigid time lines and course production schedules.

In order to see what happens in the life of a "typical CAI author" once the initial courseware development has been completed, let's make three different visits with the typical author. The first visit takes place immediately after the last line of computer code for the new course has been entered; the second visit takes place during the first semester in which the course is being used by students enrolled in a credit course; and the third visit takes place at the end of the semester when the external evaluator from the granting agency requests course data from the author.

VISIT 1: That's it! It's complete! I never thought we would make it in time for the next semester. Oh, I know that the "good book" on course design recommends that the course should first be used with a trial group of students. That's not a bad idea, but first I'll have to schedule some student terminal time.

Scheduling of terminals? I forgot all about that; it would be impossible to schedule that many terminals on such short notice.

Maybe two or three students? No, it's not worth the time and effort to track the progress of two or three students. There really is no reason to worry. The course will work without any major problems; after all, the programmer and I have been through the code a number of times. Didn't I design the material? It can't fail. If any minor problems arise the students will have to grin and bear it.

VISIT 2: How did all those spelling and grammatical errors get into the code. Those students are certainly picky, especially that little elementary school English teacher.

Branching errors! I didn't know there were that many branches in the entire course. Why should the students be so upset? It doesn't seem like a bad idea to cover some of the material four or five times, even if they didn't ask for the review. Drill and practice doesn't hurt. Imagine how much longer they will retain the material with those added reviews.

How am I supposed to keep everyone happy? Now some of the students who were accidentally branched past some of the important course material are asking to be moved back to those portions of the course that were missed. Maybe a facility which permits students to move about in the course will be available at a later date; I hope so. For now, however, they should sign-off and return tomorrow. Hopefully the systems analyst or the computer operator will be able to reposition these students at the correct location before they return to continue with the course tomorrow. I'll have to make a note in my calander to stay away tomorrow. Lots of luck!

He says he found an error in the answer analysis? That's just not possible. He has the audacity to say he entered a correct answer to the question, but that the system responded by displaying a message that he was wrong. He probably can't type. I'll enter the answer for him. What? That's just not

possible! There must be an error in the computer code. I'd better figure out some way to have him bypass this section of the code.

How in the world am I supposed to figure out where this error is in the course code? Just one minute! Even if I find the portion of the course code that is in error, what good will that do for the students who are on-line now? The instructional system does not permit authors to make dynamic changes to the course code while students are signed-on to the course. Changes can only be made when the students have signed-off the course.

Damn. It would have been much easier not to use CAI as a mode of instruction.

VISIT 3: How in the world am I supposed to find all these data for the external evaluator. I know that the system automatically records a lot of the students' performance data, but the system report generators do not provide the information requested. I know, I'll just ask the computer operators to make a dump of the raw performance data similar to the data I am provided with. The pile of computer paper will measure at least four or five feet in height. That should keep him busy for at least one year.

He will probably ask me about proposed course modifications. Well I have some written evaluations which were made by the students, a student attitude questionnaire, and some post-test performance data.

Perhaps I won't make any changes this year. If I start right now, I just might be able to complete the development of three new chapters for the course prior to the fall semester. I can hardly wait to get started!

The above soliloquy likely describes the feelings of many first-time CAI authors and, I would guess, many first-time CAI "courseware design teams". Without adequate quality control of the courseware

being developed and the provision for the allotment of adequate time and personnel to revise portions of the course, the problems that might occur when students begin using the course could be insurmountable.

LACK OF CLEAR DIRECTIONS

Where does the CAI author turn for help in examining techniques that give one assurance that the course being designed has a high probability of being successful? Unfortunately, the literature in the field of CAI tends to focus upon topics such as CAI hardware and software requirements, comparisons of different authoring languages, descriptions of courses that are being developed or are already being used at various educational institutions, types of student terminals required for different modes of CAI, etc. In addition, there have been many articles describing studies in which some combinations of different CAI modes and conventional instruction were compared. But where are the articles relating the experiences of authors who have undertaken systematic approaches to diagnose errors in the courseware by using the monitoring facilities available to them through the computing system which delivers the instruction? Where are the detailed descriptions of various strategies used to optimize the courseware? What were the successes or failures of the various optimization procedures attempted? One must search far and wide to find enough material to constructively help the CAI author.

It appears that most CAI authors are caught in the syndrome of continuing the production of new courseware while living with the first version of their courseware rather than taking time to modify and

improve some of the courseware they already have designed. For example, many of the early CAI studies in the 1960's were concerned with "proving" that CAI was more efficient and produced better post-test results than did conventional instruction. Little has been done to take some of these early CAI programs and pass them through a series of diagnosis and review (optimization) cycles and then replicate the studies. Outside of the drill and practice materials produced at the Institute for Mathematical Studies in the Social Sciences at Stanford University, few of the early CAI programs have had much of an impact in terms of number of students using the programs and the number of institutions at which versions of the programs are being used.

The need for review and evaluation procedures is brought home by Steinberg (1977). In her review of studies that employed the concept of learner control in conjunction with CAI, Steinberg found no conclusive evidence to support or reject the use of learner control features in a CAI program.¹ For example, Seidel et al. (1975) found that some students used learner control effectively, while other students used poor learning strategies and would perform best if their

¹The TICCIT System (Stetten, 1972) was designed with a set of learner control keys on the student keyboard and the initial courseware designed for use on the TICCIT System incorporated the learner control features (Bunderson, 1973a and 1973b). It is interesting to note that the TICCIT group took a risky step in generating a great deal of courseware based upon the concept of learner control at a time when there was still little agreement as to the effectiveness of learner control. During the formative stages of TICCIT the educational goals for the project were described to be humanization, relevance, and individualization (Stetten, 1971).

movements through a course were controlled by the computer program.²

The reasons for the conflicting results in the learner control studies likely are due to the many interpretations of learner control and to the methods used to implement these interpretations in learner control studies. Because many of the learner control studies were based, in part, upon the generalized results of "aptitude by treatment interaction" there tended to be two major variables being manipulated in the learner control studies. The two major variables manipulated were the amount of learner control over course flow (ranging from complete student control to complete computer control over instructional paradigm selection, topic sequencing, selection of objectives, amount of practice, amount of review, and difficulty level) and learner characteristics (motivation, aptitude, anxiety, reasoning ability, and inquisitiveness). Much of the recent work in this area has attempted to formulate generalized guidelines to help instructional designers produce a better product which more adequately meets the needs of individual students. The results are confusing. Obviously more research is needed before some general directions will evolve. In examining the various learner control studies, it appears, from an instructional design perspective, that the concept of course optimization has been all but overlooked. One must wonder whether any of

²If future follow-up studies of Seidel's study confirm the finding that low aspiration students (who pay the same fees as high aspiration students) should not be permitted access to the same learner control features permitted high aspiration students, and this then becomes policy on a large scale CAI system in the public school sector, for example, the legal implications (in terms of human rights) would indeed be interesting.

the studies would have produced different results if the CAI courseware had been instructionally optimized, based upon data from student performance files and evaluations of the courseware by the authors' peers?

EVALUATION CYCLE

Because of the conflicting results of the research data in the field of CAI, in particular the results of studies of aptitude by treatment interaction and learner control, there appears to be a need for CAI authors to become conversant with the recent writings in educational evaluation.³ However, the reader should be cautioned that the readings in educational evaluation are almost as varied in their directions as the directions of studies involving CAI. This is due, in part, to the fact that much of the recent activity in the area of educational evaluation began as late as the mid-sixties.⁴

Two articles by Dick (1977a and 1977b) provide the CAI author with an excellent overview of procedures that can be followed in evaluating his courseware. The evaluation process is separated into two components, formative and summative. Formative evaluation emphasizes "thorough testing" of courseware prior to regular use by students. The purpose of formative evaluation is to provide the author with as

³Some of the more outstanding readings in this field are by Popham (1975), Scriven (1975) and Stufflebeam (1971).

⁴Much of the credit for the growth of interest in educational evaluation can be attributed to the essays by Scriven (1967) and Stake (1967).

much information as possible to diagnose, revise, and strengthen the courseware while it is still in the development stage. During this stage, the author attempts to obtain course performance data by a systematic tryout of the courseware with students. Dick lists one-to-one, small group, and field trial as three phases of formative evaluation.

In contrast to formative evaluation, summative evaluation is intended to provide evaluative information to the decision-makers who must make long term decisions regarding the project. Emphasis is upon the broad outcomes of the project. Summative evaluation is usually conducted at the end of a project by an evaluator who is external to the project, whereas the formative evaluation is often conducted by project staff.

The formative evaluation process should be of paramount concern to the CAI author. Much of the success in receiving a positive summative evaluation will likely be a direct reflection of the extent to which the author exposes the courseware to rigorous formative evaluation.

How does the CAI author ensure that the courseware is adequately evaluated and optimized during the development phase of the project? One approach is to plan a thorough testing procedure for the courseware. If anything is to go wrong with the course, it's advisable to find the errors and make the modifications before students get to use it.

The following section examines two phases of the development cycle, the pre-coding phase and the coding phase. Procedures that CAI authors may wish to consider in the optimization of courseware are

included in the two phases.

Pre-coding Phase:

Although this phase often requires less time than the coding phase, it is probably the more crucial of the two. In this phase, each lesson script is reviewed and evaluated by the author's peers. An excellent illustration of how this approach has been used, was documented by Call-Himwich (1977). In an attempt to formalize techniques to help CAI authors write instructionally effective materials, Call-Himwich examined the concept of using the authors' peers as reviewers at various stages of the development process as one method of providing feedback to authors regarding the instructional adequacy of the courseware. The peer feedback formed a basis for optimizing the scripts prior to having the course coded and entered into the CAI system. One finding was that the direction of the review should be agreed upon prior to the review by the author and reviewer.

By providing evaluative feedback at an early stage of the coding process, authors tend to be less "ego involved" and thus more receptive to peer comments than they would be after the material becomes operational on the computing system. Call-Himwich found that authors were more positive to evaluations conducted by peers who were perceived to be at the same authority (status) level as the authors. In addition, authors were more receptive to comments made by peers who were regarded, as "experts" in instructional design rather than peers who were regarded as subject matter specialists. Perceived of most use to authors were evaluative comments which dealt with possible alternative instructional strategies and changes suggested to rectify grammatical errors. However, it should be noted that the author-

reviewer approach may not be acceptable to some CAI authors.

Call-Himwich also found that review procedures worked best if the course reviewers were on-site. She found that author and review teams worked together best when they were able to communicate on a one-to-one basis. There was a tendency, for example, to have written comments from external reviewers filed away and no action taken.

Since courseware design generally tends, over time, to be regarded as a personal product, it is important for the reviewer to be a person who has a "flexible adaptive personality". Thus, if you are in charge of a CAI courseware production project and you intend to use peer reviewers, be certain that the reviewers and authors are compatible; a mismatch between author and reviewer could negate this phase of the evaluation process.

Coding Phase:

Once the courseware has been coded, the material can then be evaluated in a number of different ways. The evaluation process will again involve the author's peers, as well as analyses of student performance data, student attitudes and computer support programs.

The length of time required for this phase is variable. Prior to exposing the courseware to a final summative evaluation, the author should ensure that at least two or three intact groups of students make separate passes through the courseware. After each pass through the course by a group of students, there will necessarily be review, diagnosis, and revision of the courseware. The total number of student groups the author permits to pass through the courseware before a point is reached where further modification is unnecessary will vary with each author and his priorities. Once the course

reaches a static stage, as far as further revisions and changes are concerned, the courseware is then ready for a final pass by a group of students and the summative evaluation can be conducted.

Many of the author support facilities described in this phase are already provided by some of the current CAI systems. The extent to which computer assistance is available to the author, of course, depends a great deal upon the CAI system being used and the number of application programs available.

Initial Computer Support Programs:

From an author's perspective, particularly from a time consumption point of view, it is important to provide the author with automatic facilities for sorting, collating, and text editing of course code and student performance data. The facilities provided for the author will depend upon the application programs available on the user's CAI system. For example, Romaniuk and Montgomery (1976b) described an author support system which accepts as input course code from an IBM 1500 System, reformats the code on an Amdahl/470 computing system, and produces a hard-copy listing of all CRT displays at each question node and the associated anticipated answers which were supplied by the author. This means an author does not require access to a computer terminal for some segments of the evaluation phase. This is particularly important at an installation where there is a scarcity of available computer terminals and the author is placed in a position where he must compete with students, authors, and systems programming staff for terminal time. By having access to a hard-copy display of the course, the author (or reviewer for that matter) can quickly browse through the hard-copy material and spot grammatical and spelling errors, as well as errors in the types of student

responses anticipated by the author. The errors can be quickly noted on the hard-copy and a request for code changes can be passed on to a coder or programmer.

Automatic support programs can be extremely valuable aids in assisting the author to quickly and efficiently "check out" courseware prior to use by reviewers and "test students". Included in these computer support programs should be a facility to document portions of the courseware directly from the course code. Because CAI authors tend to vary in their eagerness to document their courseware, some of this process can be easily accomplished by computer support programs. For example, such programs would provide authors with cross-reference tables containing all course variables that can be extracted from the course code (e.g., what are all the occurrences and locations of commands referring to the display of slide 16, modification of status indicator 12, questions with a student time-out latency of less than 20 seconds, etc.,). There are, however, some components of the documentation process which rely upon the author for information. This portion of the documentation process would perhaps be best left to an interactive authoring system which would query the author for information. For example, the CAI system would quiz the author for further information when the author enters an instruction which references an audio-visual terminal component, defines a new variable, generates a new graphic, establishes an examination subsystem, adds specialized answer analysis routines, manipulates added student performance data, initiates a new concept block or defines a new course lesson.

Initially, the documentation process can be time consuming,

but the time spent can be invaluable, especially when the author enters a series of revision cycles and numerous portions of the course undergo changes. However, imagine the case of the author who does not document his course code (you probably know one or two). During the course modification process, the author modifies a number of sections of code, including complete removal of references to a couple of variables which are used throughout the course (the author couldn't remember why they were there and thought they were not needed). The same author then makes similar changes to a number of other variables which are also used in various portions of the course. When the test students finally access the course, the result is likely to be nowhere near the creative instruction imagined by the author; the result probably more closely resembles instructional chaos.

Reviewers:

As in the coding phase, the author's peers can again be requested to review completed sections of code. It is recommended that the reviewers examine short course modules rather than wait until the entire course has been programmed. This will help ensure that comments are returned to the authors in short order and major changes undertaken before too much effort has been made in coding large sections of the course.

Depending upon the CAI system used, the approaches used in the review process will vary. According to Call-Himwich (1977) the best approach still appears to be one in which the author and the reviewer sit at a terminal and interact on a one-to-one basis. Call-Himwich also reported that the "master-slave" terminal approach was used when the reviewer and author were located at different PLATO terminal

sites. The master-slave approach is one in which the reviewer executes through the course while the author sees a duplicate copy of the reviewer's screen display. Then, by using the communication feature available on the PLATO system, the author and reviewer communicate by typing short messages which are displayed on both display screens.

Another approach is to have the reviewer view a course alone and, whenever he wishes to make comments, to direct comments to the author's "mailbox". The reviewer's comments will accumulate in the author's "mailbox" until the author next signs on to the course. At that time, the author will then be notified that messages are available for him in the "mailbox". If such a facility is available on a CAI system, there should be a provision for the system to automatically include the sender's name, date, time, and location in the course at which the comment was made (this same feature can then also be used by students). Authors should be required to reply to comments. Requiring the author to reply to comments might reduce the probability that comments will simply be "filed".

Single Student Test:

One of the most important tests of the adequacy of the courseware occurs when "test students" begin using the course on a trial basis. After making the initial first-pass modifications to a course module (or even a portion of a module), the author may wish to have individual students work through the module. This will actually be the first use of the course by students for whom the course was targeted. If any audio-visual materials are to be used in the course, the materials will likely be in preliminary draft form. Rough

sketches of future slides and printed audio scripts would be used rather than final computer ready materials.

The author should sit at a terminal with a single student. The intent is to carefully watch for any problems the student has with the courseware and to ask questions of the student when it appears the student's responses are quite different from those anticipated by the author. Emphasis is upon noting possible instructional problem areas in the courseware, spelling, grammatical and text formatting errors, the appropriateness of the audio-visual materials, and tentative suggestions for modifications to the courseware. Any tests or questionnaires to be used in the course should be included in this phase.

Small Group Test:

After making revisions to the courseware, based upon the cumulative findings of having a few individual students review the course, the course is then ready for use by a small group of students. A group of five to twelve students would be sufficient in any single session. The number of students should be kept small enough for the author to be able to move about the terminal room and assist students when they request help or clarification of the courseware. During this phase, the audio-visual materials should either be in, or near, finished form for full implementation on the computer system.

The computer facility should ease the record-taking task of the author in this phase by permitting each student to access a "comment" routine at any point in the course. When an error is encountered, the student would merely access the comment routine and type a comment. After typing the comment, the student would automatically be returned to the portion of the course at which the comment

was made. The CAI system would attach the student's name, location in the course, date, and time to each comment and store the comments in the author's "comment file" (mailbox).

Simple changes to text, such as spelling, grammar, and formatting errors should be made as soon as possible. This would give the author a chance to see what effect the changes make on any of the students who have not yet accessed that portion of the course. More complex changes should be held in abeyance until the author has a chance to view cumulative student performance data.

The tests and questionnaires should be computerized and included in this phase of the evaluation cycle. When the students complete certain modules, or the entire course for that matter, the author may wish to interview the students individually or as a group to discuss their general impressions of the course.

Class Test:

Once the various course modules have all been tested by small groups of students and all revisions made, the courseware is ready for a test by a "class" of students. The students should, of course, be from the target audience for whom the courseware was written.

Dick (1977a) indicated that at this stage the author could release the material to another instructor and then maintain a low profile during this test phase. However, this decision is the responsibility of the author. He should only release the courseware if he is quite certain that the course has been fully tested and optimized. It may be best for the author to be in charge of the first class of students to test the course. Although it is true the author has already permitted a number of students access to the course, it is

not surprising, especially if the course is 30-40 hours in length, to have numerous sections of the course that have not been exposed to use by students.

During this test phase, the end of module tests and attitude questionnaires will be of primary interest to the author. Did the students achieve the levels of learning projected and were their attitudes toward the courseware positive? If the students' test scores are consistently below the levels anticipated, there would need to be a complete review of the course and the available data to see if the course should be completely revised or simply scrapped.

The attitude data can often indicate portions of the course in need of revision not identified by examination of student test scores. For example, Hunka, Romaniuk, and Maguire (1976) used attitude questionnaire data as a basis for program modification in a basic statistics course. One note of caution, if you include an attitude questionnaire in conjunction with an end of module test, place the attitude questionnaire prior to the test (for obvious reasons).

Field Test:

At this stage, the courseware is ready for distribution to instructors at other institutions and/or released to instructors at the author's institution. The author should ensure that the instructors try out the course prior to use by their students. In addition, the instructors should be present during regularly scheduled student sessions and should complete weekly evaluation forms indicating potential problem areas in the course. In order to effectively help the students through any portions of the course still in need of revision,

the instructors must be conversant with the instructional strategies, general methods of questioning, amounts of student control over course sequencing, and the available review facilities. The instructors must also be shown how to access the student performance records to enable them to review examination results with students.

All student auxiliary materials must be in a final form at this time. The author should ensure that procedures to record student performance data are in effect and that the data can be released to the author for use in diagnosis of student performance. Once this phase is complete and the necessary revisions made, the author is then in a position to make the course available for summative evaluation.

DIAGNOSIS AND REVISION OF COURSEWARE

This component of the instructional design and implementation process can often be the most difficult and time consuming. Activities associated with diagnosis and revision begin at the time the author starts scripting the course and increases in intensity as the evaluation cycle is initiated.

Although a number of investigators have recommended different specific procedures for revising instructional materials, there is little support from recent studies for the generalizability of revision procedures (Gropper, 1975). Gropper indicated that more emphasis must be placed upon diagnosis (described as the mirror of task analysis) if authors are to more effectively modify instructional materials. If a student fails to attain certain skills, the author must be able to analytically trace the student's performance through a section of the

course and isolate missing instructional treatments and/or inappropriate sequences of materials. This procedure may also be conducted with group data. The book by Gropper (1975) is an excellent source of information for the beginning CAI author who wishes to become aware of tools and techniques that can be applied to the diagnosis and revision of instructional materials.

In order to adequately diagnose student data the author often needs access to a variety of student performance reports. Since the task of collating and sorting vast amounts of student data is a tedious undertaking, it is most likely not done unless computer support programs are available. The success of the diagnosis process depends upon being able to quickly examine test and response data.

There are a number of report generators available on some of the current CAI systems (e.g., IIS and PLATO). An excellent example of the use of a report generator to provide an author with a summary of student response data from a drill and practice program was reported by Hofstetter (1975). If report generators are to be used by authors, however, the procedures required to initiate the generators must be simple to use. An author shouldn't be an expert in job control languages and a programming language in order to access student response data. Montgomerie and Romaniuk (1976b) described an interactive system that permits authors to specify the types of reports they wish to have generated. Techniques were devised by the Division of Educational Research Services (DERS), University of Alberta, to assist authors to easily request tailored student performance reports.⁵

⁵DERS has operated an IBM 1500 Instructional Computing System for the past decade. The system currently supports 20 student stations.

The need for quick and easy access to student performance data grew more acute as longer courses were developed, more students were accessing courses, and the authors expressed an increasing need for rapid turn-around of performance data for courseware revision purposes. For example, the basic statistics course described by Hunka et al. (1976) requires an average of 70 hours to complete; a class of 35 students generate approximately 125 thousand student records (each student record consists of 360 characters of information). Because many of the CAI authors at DERS have little familiarity with computer languages, a series of CAI support programs were designed and implemented in interactive mode on an Amdahl/470 (which uses the Michigan Terminal System). The interactive program optionally explains the use of the available support programs, queries the author about a series of tailoring options that are available with each program, checks the user's input for errors and prompts the user for legitimate input, estimates the parameters for the "sign-on" card, and issues a call to run the program. The author does not need any knowledge of topics such as job control language, tape labels, blocking factors, format of performance data, or sort parameters. A typical request by an author for a summary of the performance data may take five minutes or less and cost approximately sixty cents; the result is a program consisting of 150 statements (similar to job control statements) which runs nine mainline programs, issues six sort calls and mounts up to five tapes.

The stages of development of one of the main CAI support programs used at DERS was documented by Romaniuk and Montgomerie (1976a). The program provided CAI authors with the following three summaries

for any specified course segment:

- a) frequency counts of the number of responses in each of the answer categories for each question asked in the course (for a specified group or for individual students).
- b) student comments (listed by course location).
- c) cumulative data for each student regarding the time-on, number of correct responses, number of responses, number of time-outs, date started, date completed, and group data regarding time on the segment (e.g., mean, standard deviation, and range).

An illustration of the output from the group performance generator is displayed in figures 1-3.

Figure 1 was obtained by merging the course source code and the student response data, and recreating the CRT associated with each question encountered in the course. The various answer categories and the answers anticipated by the author for that question are listed horizontally below the CRT display while the cumulative frequencies of students who selected each answer category on trial 1, 2, 3, 4 and more than four are listed vertically. To examine the performance of the courseware at each question, the author merely scans the cumulative frequency data for an inordinate proportion of unanticipated responses. To see the actual student responses for that question, the author merely scans the responses (listed in ascending order by student number) in the various unanticipated answer categories. Each student response is prefaced by the student's number and the sequence number of that response for the student at that question.

By using the group performance analysis program, a DERS CAI author can quickly detect questions that appear to be causing students difficulty, scan the unanticipated answers looking for groups of

similar responses, and include these groups in the block of anticipated responses. Diagnosis of the student responses will often indicate tactical errors in the design of some of the preceding instructional material. After courseware revisions have been made, the authors can save the summary performance listings for comparison with the summary performance data from the next group to access the course. Comparison of both course versions gives the author a rough measure of any improvement in the courseware. Hopefully, the new version of the course will result in increased learning, in less time, and with an accompanying increase in positive students attitudes toward the course.

One criticism of the diagnosis and review approach used at DERS was the tendency for authors to use a micro-diagnosis in making decisions about revision of courseware. Gropper (1975), for example, recommended that instructional designers increase emphasis of macro-analysis in the process of diagnosing courseware deficiencies; looking at a series of isolated responses made by a student can be frustrating for the author because reasons for the pattern of student responses are often quite baffling. Brown et al. (1977) provided an excellent illustration of this dilemma for teachers who used part of an available mathematics drill and practice program. They investigated the types of responses made by students in responding to typical questions generated during drill and practice sessions. Of particular interest was the occurrence of similar errors in different skill areas. The position taken by Brown et al. was that the errors were not random but, in fact, were the result of a consistent algorithm (albeit wrong) used by the student in solving the

problems. One conclusion is that perhaps we should be looking more for a consistent pattern of incorrect responses made by students as they pass through a series of review exercises and attempt to find ways to correct the students' algorithms that are in error.

SUMMARY

Formative evaluation of courseware is a vital step for the CAI author. Without rigorous evaluation during the development cycle, diagnosis and revision of courseware will likely be haphazard and inefficient, and the probability of the course being a success quite low.

The fact that a computing system is used in the instructional process immediately provides the author with an advantage during the formative evaluation process that is not generally available in other forms of instruction. Student responses and other relevant data can be automatically saved and then potentially manipulated and output in almost any form desired by the author. If author support programs are available on the user's computing system, the time required for the author to wait for the output of summary student performance reports can be minimal. This means an instructional module can be tested by students and the author can be in possession of summary reports which detail the students' performance on post-tests and summarize student attitudes toward items related to the appropriateness of the subject matter, relevance and variety of the examples, and difficulty of test items. In addition, the author can examine within-program student responses for error patterns. The author can make changes to the module, permit another group of students to take the module the next

day, obtain another set of student performance reports, and then compare the results of the current and past versions of the courseware.

Another exciting feature for the CAI author is the potential to sit at a terminal and trace a student's path through the course. This feature could be useful in assisting students when instruction breaks down or when the author is searching for possible explanations to a series of inappropriate responses by a student.

As CAI authors continue to search for more sophisticated procedures to automatically obtain succinct and meaningful reports of student and courseware performance, a more rigorous framework for formative evaluation procedures of CAI courseware will hopefully evolve. Until then, however, authors need to pay particular attention to systematic analysis of student performance prior to any revision of courseware. The number and extent of the revisions will, of course, depend upon the situation. Regardless of the number of program goals achieved by the courseware, the author must continue diagnosis and revision until the courseware fulfills the requirement - that students perform successfully on the final post-instruction tests.

REFERENCES

- BAKER, E.L., and ALKIN, M.C. Formative evaluation of instructional development. Audio-Visual Communication Review, 1973, 21 (4), 389-418.
- BROWN, J.S., BURTON, R.R., and LARKIN, K.M. Representing and using procedural bugs for educational purposes. A paper presented at the 1977 ACM Conference, Seattle.
- BUNDERSON, A.V. Team production of learner controlled courseware: A progress report. ICUE Technical Report #1, Provo: Brigham Young University, May, 1973a.
- BUNDERSON, A.V. The TICCIT project: Design strategy for educational innovation. ICUE Technical Report #4, Provo: Brigham Young University, September, 1973b.
- CALL-HIMWICH, E. An assessment of lesson review as a formative evaluation tool. MTC Report #8, Urbana, Ill.: University of Illinois, Computer-Based Education Research Laboratory, January, 1977.
- DICK, W. Formative evaluation. In L.J. Briggs (Ed.), Instructional Design, Englewood Cliffs: Educational Technology Publications, 1977a.
- DICK, W. Summative evaluation. In L.J. Briggs (Ed.), Instructional Design, Englewood Cliffs: Educational Technology Publications, 1977b.
- FLETCHER, J.D., and SUPPES, P. The Stanford project on computer-assisted instruction for hearing-impaired students. Journal of Computer-Based Instruction, 1976, 3 (1), 1-12.
- GROPPER, G.L. What should a theory of instruction concern itself with? Educational Technology, 1976, 16 (10), 7-12.
- GROPPER, G.L. Diagnosis and revision in the development of instructional materials. Englewood Cliffs: Educational Technology Publications, 1975.
- HOFSTETTER, F.T. GUIDO: An interactive computer-based system for improvement of instruction and research in ear-training. Journal of Computer-Based Instruction, 1975, 1 (4), 100-106.
- HUNKA, S., ROMANIUK, E.W., and MAGUIRE, T.O. Report on the use of the computer assisted course STAT1. RIR-76-7, Edmonton, Alberta: University of Alberta, Division of Educational Research Services, November, 1976.

- POPHAM, J.W. Educational evaluation. Englewood Cliffs: Prentice-Hall, 1975.
- ROMANIUK, E.W., and MONTGOMERIE, T.C. After implementing your course-- What's next? Proceedings of the second Canadian symposium on instructional technology. Quebec City: National Research Council, June, 1976a, 377-396.
- ROMANIUK, E.W., and MONTGOMERIE, T.C. Interactive programs: A simple way to allow naive users to manipulate extensive data bases. Proceedings of the University of Alberta Users Applications Symposium. Edmonton, Alberta: University of Alberta, May, 1976b, 121-144.
- SCRIVEN, M. Evaluation perspectives and procedures. In Popham, W.J. (Ed.), Evaluation in Education: Current Applications, Berkeley: McCutchan Publishing Company, 1974.
- SCRIVEN, M. The methodology of evaluation. In R.E. Stake (Ed.), Perspectives of Curriculum Evaluation. American Educational Research Association Monograph Series on Evaluation, No. 1, Chicago: Rand McNally, 1967.
- SEIDEL, R.J., WAGNER, H., ROSENBLATT, R.D., HILLELSOHN, M.J., and STELZER, J. Learner control of instructional sequencing within an adaptive tutorial CAI environment. Technical Report 75-7, Alexandria, Va.: Human Resources Research Organization, June, 1975.
- STAKE, R.E. The countenance of educational evaluation. Teachers College Record, 1967, 68, 523-540.
- STEINBERG, E.R. Review of student control in computer-assisted instruction. Journal of Computer-Based Instruction, 1977, 3 (3), 84-90.
- STETTEN, K.J. An overview of the TICCIT program. McLean Virginia: MITRE Corporation, June, 1972.
- STETTEN, K.J. The technology of small, local facilities for instructional use. In Levein, R.E. (Ed.), Computers in instruction: Their future for higher education, Santa Monica, Ca.: Rand Corporation, July, 1971.
- STUFFLEBEAM, D.L. The Relevance of the CIPP evaluation model for educational accountability. Journal of Educational Research and Development, 1971, 5 (1).

stat1-019

PAGE: 19

P IDENTIFICATION = 190012-010

RT DISPLAY : EXAMPLE: You wish to throw a die 180
times to see if it can be
classified as unbiased.
1) First assume the null hypothesis
(i.e., that the die is unbiased)?
If the die is unbiased, what frequency
of throws would you expect in each
of the score categories?

HAID	DESCRIPTION OF STUDENT RESPONSE		ATTEMPT:	1	2	3	4	5	SUM
ot				0	0	0	0	0	0
c1	30	\$ NUMERICAL ANSWER \$ 30		17	17	2	3	0	39
v2	6	\$ NUMERICAL ANSWER \$ 6		0	0	0	0	0	0
u3				16	0	2	0	0	18
u4				0	1	0	0	0	1
u5				0	0	1	0	0	1
TOTALS:				33	18	5	3	0	59

KEYBOARD RESPONSES TO HAID 'u3'

p151(1) 1. 90
p154(1) 2. 90
p156(1) 3. 90
p162(1) 4. 90
p163(1) 5. 90
p164(1) 6. 90
p166(1) 7. 90
p168(1) 8. 3
p171(1) 9. 1/6
p172(1) 10. 90
p173(1) 11. .50
p175(3) 12. 100
p179(1) 13. 90
p182(1) 14. 90
p182(3) 15. 90
p185(1) 16. 90
p187(1) 17. 180
p217(1) 18.

KEYBOARD RESPONSES TO HAID 'u5'

p217(2) 1.

KEYBOARD RESPONSES TO HAID 'u2'

p217(3) 1.

FIGURE 1. Sample of summary report of within-program student
response data.

COURSE: stat1SEGMENT: 19

LISTED: FEB 22, 1978

STUD. NO.	TIME (HR:MIN)	CORR.	RATIO	STARTED	FINISHED
p034	0:00	0	0.00	MAY 25, 1977	MAY 25, 1977
p150	1:21	23	0.28	JUL 14, 1977	JUL 14, 1977
p151	2:10	23	0.17	AUG 04, 1977	AUG 04, 1977
p152	2:08	26	0.20	AUG 02, 1977	AUG 03, 1977
p154	1:41	23	0.22	AUG 05, 1977	AUG 05, 1977
p155	1:10	24	0.34	AUG 01, 1977	AUG 02, 1977
p156	1:42	23	0.22	JUL 26, 1977	JUL 27, 1977
p157	1:15	23	0.30	AUG 03, 1977	AUG 04, 1977
p158	0:54	23	0.42	JUL 13, 1977	JUL 15, 1977
p159	2:12	25	0.18	JUL 30, 1977	AUG 01, 1977
p161	0:58	29	0.50	JUL 25, 1977	JUL 26, 1977
p162	1:25	23	0.27	AUG 01, 1977	AUG 02, 1977
p163	1:21	25	0.30	JUL 30, 1977	JUL 30, 1977
p164	0:40	23	0.57	AUG 02, 1977	AUG 03, 1977
p165	2:18	49	0.35	JUL 28, 1977	AUG 01, 1977
p166	1:51	23	0.20	AUG 01, 1977	AUG 01, 1977
p167	1:58	24	0.20	AUG 03, 1977	AUG 03, 1977
p168	1:19	23	0.29	AUG 02, 1977	AUG 02, 1977
p171	2:35	49	0.31	JUL 28, 1977	AUG 02, 1977
p172	1:48	23	0.21	AUG 06, 1977	AUG 06, 1977
p173	0:56	23	0.41	AUG 04, 1977	AUG 04, 1977
p175	2:25	62	0.42	AUG 06, 1977	AUG 10, 1977
p177	1:15	24	0.32	AUG 02, 1977	AUG 05, 1977
p178	1:51	23	0.20	AUG 02, 1977	AUG 03, 1977
p179	1:52	45	0.40	JUL 29, 1977	JUL 29, 1977
p182	2:30	46	0.30	AUG 05, 1977	AUG 09, 1977
p183	1:55	23	0.20	AUG 02, 1977	AUG 04, 1977
p184	0:54	23	0.42	AUG 05, 1977	AUG 06, 1977
p185	1:59	23	0.19	AUG 03, 1977	AUG 04, 1977
p186	1:17	23	0.29	AUG 04, 1977	AUG 08, 1977
p187	0:57	23	0.40	AUG 06, 1977	AUG 06, 1977
p188	2:26	32	0.21	AUG 04, 1977	AUG 08, 1977
p189	1:24	23	0.27	AUG 09, 1977	AUG 09, 1977
p217	0:52	25	0.48	AUG 04, 1977	AUG 05, 1977

TOTAL TIME = 53:19
 NO. STUDENTS = 33
 MEAN TIME = 1:36
 STD. DEV. = 0:32
 MIN TIME = 0:40
 MAX TIME = 2:35

FIGURE 2. Student summary data.

COMMENTS FOR COURSE: stat1

DATE LISTED: FEB 22, 1978

FIRST STUDENT RECORD ON AUG 04, 1977

LAST STUDENT RECORD ON AUG 08, 1977

SEG	EPID	ID	COMMENT
019	stat1 -7	p186	At 190048-001 move your table up so you can use the calc function
	190038-49	p156	I found the f, subscript 11, very confus ing in the last frame. It should read f, subscript (1,1) for row 1, column 1.
	190048-10	p186	At 190048-001 move your table up so we can use the calc function without wiping off the table values.

FIGURE 3. Student comments to course author.